REMARKS

Applicants respectfully request reconsideration of the present application based on the foregoing amendments and the following remarks. Applicants herein amend claims 9 and 12. Claims 1-20 remain pending in the application.

Claim Rejections Under 35 U.S.C. § 112 (Second Paragraph)

Claim 9 stands rejected under 35 U.S.C. § 112 (second paragraph) as allegedly being indefinite. This claim has been amended to even further clarify the invention as originally claimed. Contrary to the Office Action, however, the claimed software program is indeed capable of generating a processor that has been configured in accordance with a user's specifications. The processor is generated by the program in the form of a hardware description, for example, a description in a language (HDL) such as Verilog. As is well known, this hardware description can then be readily used to actually fabricate the processor in silicon. The present assignee, Tensilica Inc. (www.tensilica.com), has developed highly advanced tools and commercially successful products such as Xtensa which include this capability. The present assignee has also been awarded several patents on various aspects of its technology embodied in various of its commercially successful products.

The claimed software program in claim 9 is not merely directed toward "programming" a processor, but rather "generating" a configurable processor, and then debugging software to be embedded on it. For example, the present invention set forth in claim 9 allows a developer to debug embedded software applications running on a configurable processor that has been **generated** with custom-configured instructions and state.

In view of the amendment to claim 9, as well as the above remarks, the rejection of claim 9 should be withdrawn.

Claim Rejections Under 35 U.S.C. § 102(e)

Claims 1-20 stand rejected under 35 U.S.C. § 102(e) as being anticipated by U.S. Patent No. 6,587,995 to Duboc et al. ("Duboc"). For reasons set forth more fully below, Applicants respectfully traverse this rejection.

Independent Claims 1 and 13 Patentably Define Over Duboc

Independent claims 1 and 13 both require, *inter alia*, **transmitting** a state-accessing instruction stream from a debugger to an interpreting agent, the interpreting agent then **returning** the accessed state to the debugger.

As set forth in the specification at, for example, page 2, line 3 to page 8, line 2, the present invention relates to providing embedded software development capabilities for configurable processors. One desired capability for debugging or evaluating embedded software to be run on a configurable processor is to access the state of the processor. The prior art approaches to accessing state have many problems. Accordingly, as set forth in the specification at, for example, page 10, lines 3-9, the present invention includes, for example, two aspects: a software system including a debugger that is capable of transmitting instruction sequences for processor state access to a state-access mechanism in the processor; and a state access mechanism that is capable of interpreting those sequences and returning them to the debugger.

Duboc is aimed at debugging embedded software for a processor core, but does not meet the limitations of claims 1 and 13.

First, the Office Action incorrectly attempts to read the claim limitations on various components described by Duboc. For example, claim 1 requires "transmitting, using a debugger, a state-accessing stream to an interpreting agent. . . ." However, Duboc's alleged "debugger" **includes** an "interpreter" which itself **transmits** debug parameters **to** a debug monitor. Accordingly, Duboc does not describe or suggest transmitting anything **from** a debugger **to** an interpreting agent, as required by claims 1 and 13. In fact, Duboc's **interpreter** transmits parameters **to** a monitor, which is the entirely different from what the claims require, even if, *arguendo*, Duboc's components correspond to the claimed components.

More importantly, nothing in Duboc describes or suggests transmitting a **state-accessing instruction stream** to an interpreting agent, which is then used to return processor state information to the debugging application as clearly required by claims 1 and 13. The Office Action alleges that Duboc's "debug parameters" corresponds to the claimed state-accessing instruction stream. First, however, as shown above, Duboc's debug parameters are not transmitted **to** an "interpreting agent," but are transmitted **from** an interpreter **to** a debug

monitor. Moreover, Duboc's "debug parameters" are merely used to configure a debug monitor for debugging operations. Nothing in Duboc discloses or suggests anything about including actual processor **instructions** in a "debug GUI script," much less **instructions that are executed by the processor to access state**, as required by independent claims 1 and 13.

For at least the above reasons, the rejection of claims 1 and 13, as well as claims 2-8 and 14-20 that depend therefrom, should be withdrawn.

Amended Independent Claim 9 Patentably Defines Over Duboc

Amended claim 9 requires a computer readable medium including software for **generating** a hardware description of a configurable processor, and debugger software for generating **information necessary** to describe **save and restore instructions for state** of the configurable processor **based on** the user description. According to the present specification at, for example, page 13, lines 7-17, this feature allows evaluation of embedded applications on the configurable processor to occur safely and reliably.

First, Duboc does not disclose or suggest anything about configurable processors, much less generating a hardware description of a **configurable processor** based on a user description. The Office Action points to col. 14, line 53 to col. 15, line 6 of Duboc, which passage describes a processor design flow. This design flow must be performed manually using various hardware design tools. As set forth above, the present assignee Tensilica has developed highly advanced software programs that allow a designer to specify various features of a processor (e.g. register file sizes, etc.), and then automatically generate hardware descriptions for that configured processor. These products have become well known, as is the term "configurable processor." Duboc does not disclose anything about configurable processors, much less a program that automatically generates a hardware description of a configurable processor based on a user specification.

Moreover, similarly as set forth before, Duboc does not disclose or suggest anything about **save and restore instructions** for state as specified in claim 9. Duboc merely states that "debug parameters" are transmitted to a debug monitor for debugging operations. The claimed save and restore instructions are executed by the configurable processor to save and restore state

information. Duboc does not disclose or suggest any such processor instructions, much less the claimed software library that generates information necessary to describe them based on a user description, as required by claim 9.

Still further, the cited passage in Duboc merely teaches that old code may be reused to generate new code for creating a debug monitor and debug GUI script for a specific processor core that has been designed. These "libraries" do not include instructions that are executed **by** the embedded processor, but are executed by development (EDA) tools that are used to build the debug monitor and debug GUI script for the desired processor core. Accordingly, they cannot possibly be considered equivalent to the claimed save and restore instructions.

For at least the above reasons, the rejection of amended independent claim 9 should be withdrawn.

Independent Claim 10 Patentably Defines Over Duboc

Independent claim 10 requires debugger software capable of **reading a description of save and restore state** information of a configurable processor and generating **saving and restoring state instruction streams** based on the description. This feature of the invention allows, for example, embedded application software to be debugged for a configurable processor, and then readily re-debugged for a changed configuration of the configurable processor.

As set forth above, Duboc merely describes sending "debug parameters" to a debug monitor for configuring certain debugging operations. Duboc does not disclose or suggest anything about **save and restore state instructions** that can be executed by an embedded processor to access state information, much less a debugger library that can generate such instruction streams based on a user specification of a configurable processor as required by claim 10.

For at least the above reasons, the rejection of this claim, along with claim 11 that depends therefrom, should be withdrawn.

Amended Independent Claim 12 Patentably Defines Over Duboc

Amended independent claim 12 requires an instruction-insertion server that allows retrieving system topology information of a chip containing multiple cores from a computer-readable file, determining where elements are in a system described by the file, and **directing a state-accessing instruction stream to an appropriate one** of the multiple cores in response to the determination (see the present specification at, for example, page 26, line 1 to page 31, line 3).

As set forth more fully above, Duboc does not disclose anything about state-accessing instruction streams that can be executed by a processor. Duboc further does not disclose an instruction-insertion server as required by claim 12 that directs state-accessing instructions to an appropriate core based on system topology information in a computer-readable file.

For at least the above reasons, the § 102(e) rejection of this claim should be withdrawn.

Claims 2-8, 11 and 14-20 Patentably Define Over Duboc

Claims 2-8 depend from claim 1, claim 11 depends from claim 10, and claims 14-20 depend from claim 13. Claims 1, 10 and 13 have been shown above to patentably define over Duboc. Accordingly, claims 2-8, 11 and 14-20 are patentable at least due to their dependence from patentable claims 1, 10 and 13.

The dependent claims are further patentable because they recite subject matter not disclosed or suggested by Duboc.

For example, claims 2, 6, 14 and 18 require that the interpreting agent is a monitor program. As set forth above, Duboc's alleged "interpreter" is part of the debugger, which itself transmits "debug parameters" to a monitor. Accordingly, the Office Action admits that Duboc's alleged interpreting agent and monitor are two different components, which underscores the incorrectness of the rejections. Moreover, Duboc's "monitor" does not use instructions to access state, but uses conventional debugging and simulation techniques.

Claims 3, 7, 15 and 19 require that the interpreting agent is an instruction-insertion server. Contrary to the Office Action, Duboc does not disclose or suggest an instruction-

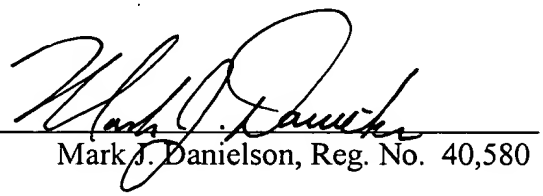
insertion server, much less that Duboc's alleged "interpreting agent" (i.e. interpreter 48) can be comprised of an instruction-insertion server.

Other dependent claims recite additional patentable subject matter. Accordingly, the rejection of the dependent claims is further improper for at least the above reasons, and the rejections should be withdrawn.

Conclusion

All objections and rejections having been addressed, and in view of the foregoing, the claims are believed to be in form for allowance, and such action is hereby solicited. If any points remain in issue which the Examiner feels may be best resolved through a personal or telephone interview, s/he is kindly requested to contact the undersigned at the telephone number listed below.

Respectfully submitted
PILLSBURY WINTHROP LLP

By 
Mark J. Danielson, Reg. No. 40,580

1600 Tysons Boulevard
McLean, Virginia 22102
Tel: 650-233-4777
Fax: 650-233-4545